# The Application of IEEE 1588 to a Distributed Motion Control System

Kendal R. Harris , Sivaram Balasubramanian, and Anatoly Moldavansky
Rockwell Automation

**Abstract**

This paper describes the application of IEEE 1588 to a distributed motion control system. Current solutions rely on proprietary implementations to time synchronize distributed motion components. With the advent of 1588, it is now possible to develop motion control solutions over standard networks such as Ethernet using commercially available technology. This paper will describe the basic operation of 1588 and motion in a working prototype.

## 1.0 Introduction

This paper describes the application of IEEE 1588 to a distributed motion control system. Current solutions rely on proprietary implementations to time synchronize distributed motion components. With the advent of 1588, it is now possible to develop motion control solutions over standard networks such as Ethernet using commercially available technology. This paper will describe the basic operation of 1588 and motion in a working prototype.

Distributed motion control applications require tight synchronization between the nodes in the system. Typically this requires the jitter between the clocks in the system to be on the order of a few microseconds. Higher performance applications are increasingly driving this requirement to the sub-microsecond range. Current solutions achieve tight synchronization between the nodes in a distributed system using proprietary networks and interface components. Custom asics in the interface cards control the distribution and synchronization of clocks throughout the system as well as timely delivery of control data.

The IEEE 1588 Precision Time Protocol provides a standard mechanism to synchronize the clocks across a distributed network. By using the 1588 protocol over a standard network a proprietary solution can then be replaced by a standard solution. Custom network interface components can then be replaced by off-the-shelf components.

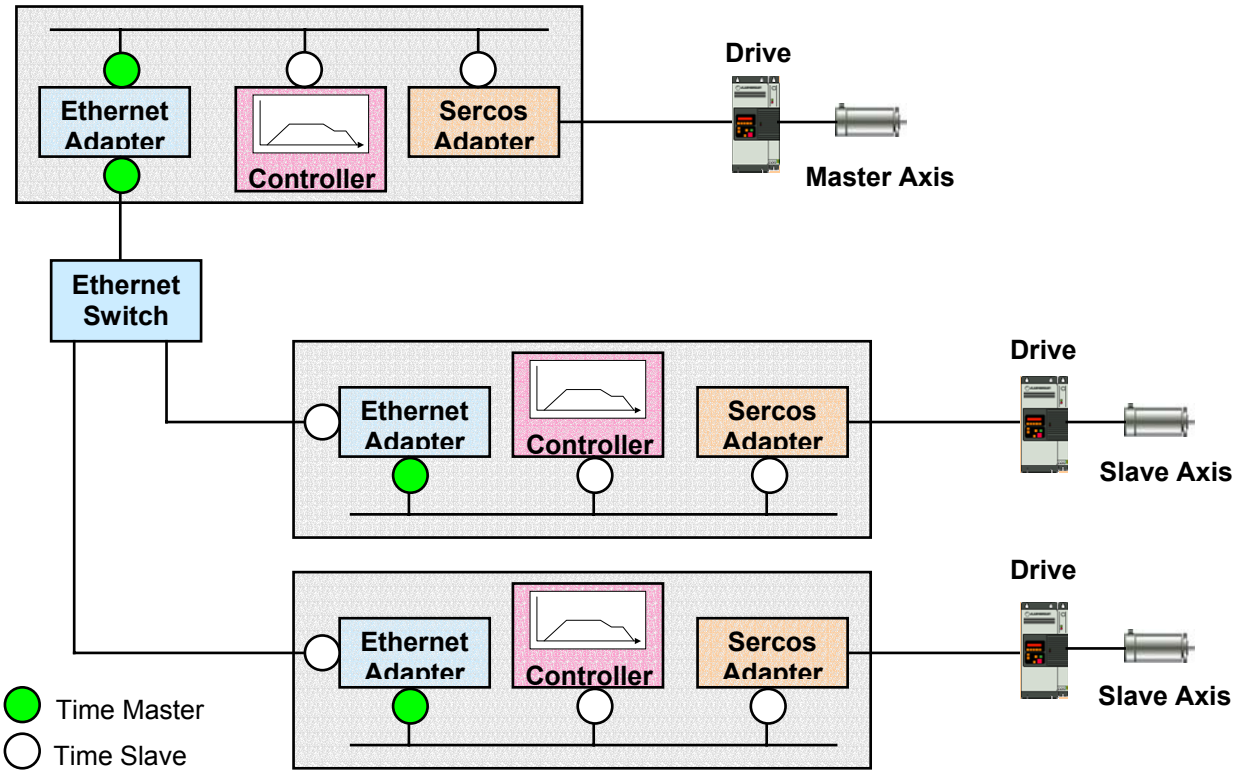
Using the 1588 protocol and Ethernet, a simple distributed motion system has been implemented to demonstrate this concept.

## 2.0 Prototype Description

The prototype motion system consists of three motion controllers. Each controller is connected to one drive over SERCOS through a SERCOS Adapter card. SERCOS is an industry standard for connecting digital drives. All the motion nodes are connected via standard Ethernet through an Ethernet adapter card.

A “motion planner” in the controller manages the position information for each drive to control motion jogging, moving, and gearing operations. Each drive is referred to as one axis of motion. One drive is the master axis and two of the drives are each slave axes. Each slave axis is geared to the master axis in a one-to-one ratio. The controller connected to the master axis on a periodic basis sends position references to each of the controllers connected to the slave axes.

The clocks on all nodes in the system are synchronized. Clock synchronization over Ethernet is accomplished using the IEEE 1588 protocol. Clock synchronization over the backplane is accomplished using a proprietary protocol that was in place prior to 1588. For both the Ethernet subnet and for the backplane one node is the subnet time master and all other nodes are time slaves.



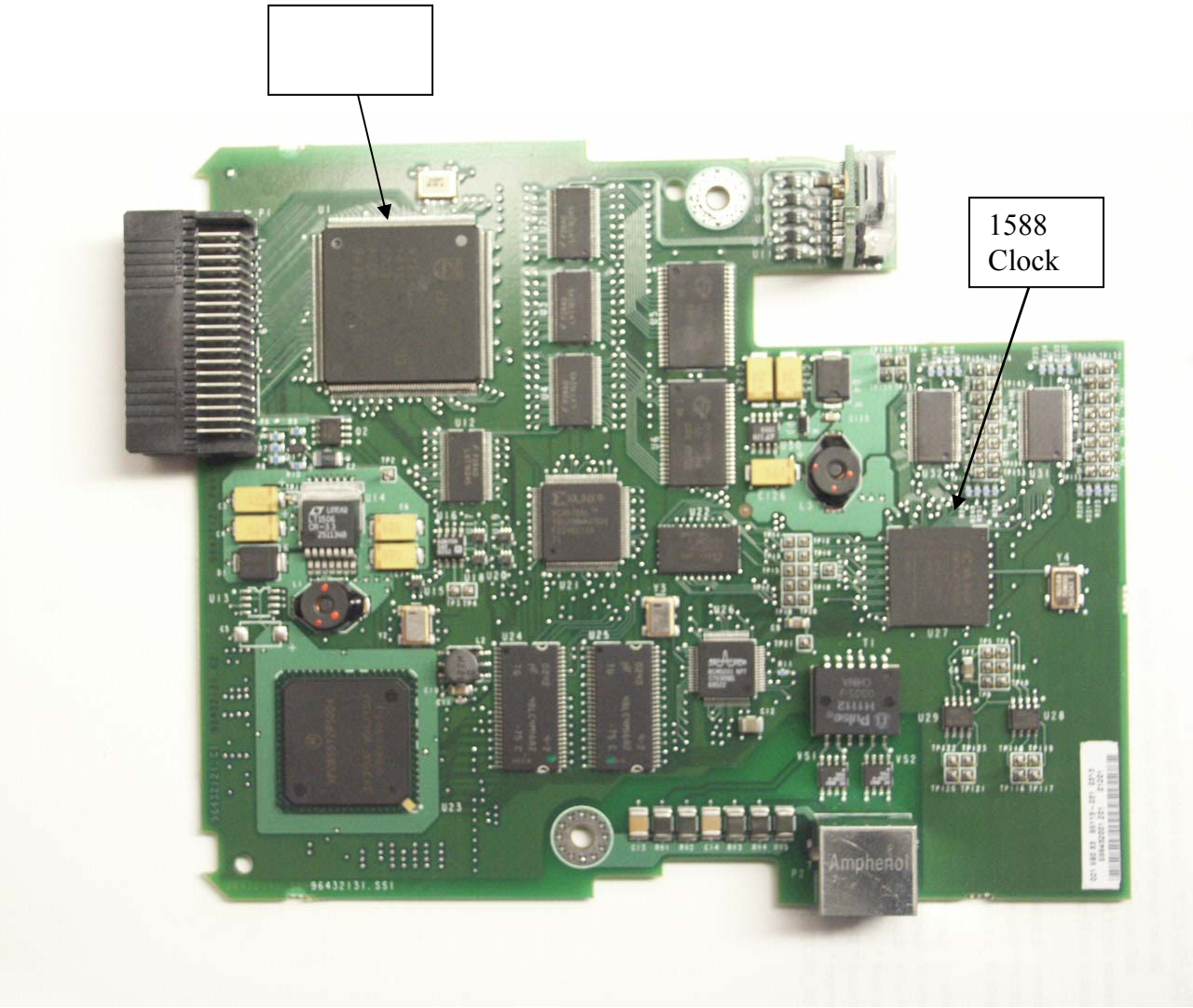

**Block Diagram**

### 3.0 System Clock Synchronization

Ethernet clock synchronization is implemented on the Ethernet adapter card. The card contains a FPGA hardware assist circuit to timestamp incoming and outgoing 1588 protocol messages. The FPGA contains a 64-bit, 25 nanosecond per tick, high resolution, tunable clock.

The 1588 protocol runs on a 50 MHz PowerPC CPU. The 1588 code interacts with the FPGA as specified by the 1588 protocol to synchronize a time slave clock to its associated master clock on the subnet. A tuning algorithm adjusts the frequency of the FPGA tunable clock once each 1588 “Sync” update cycle.

The adapter also contains an interface chip to the backplane. The clock in the backplane chip is synchronized to the 1588 clock. On the adapter the backplane interface serves as the master clock. All other clocks on the backplane are synchronized to the master clock on the adapter. A simple algorithm is used to synchronize the backplane clock to the 1588 clock. The adapter represents a 1588 boundary clock node with the backplane clock classified as a “foreign” clock.

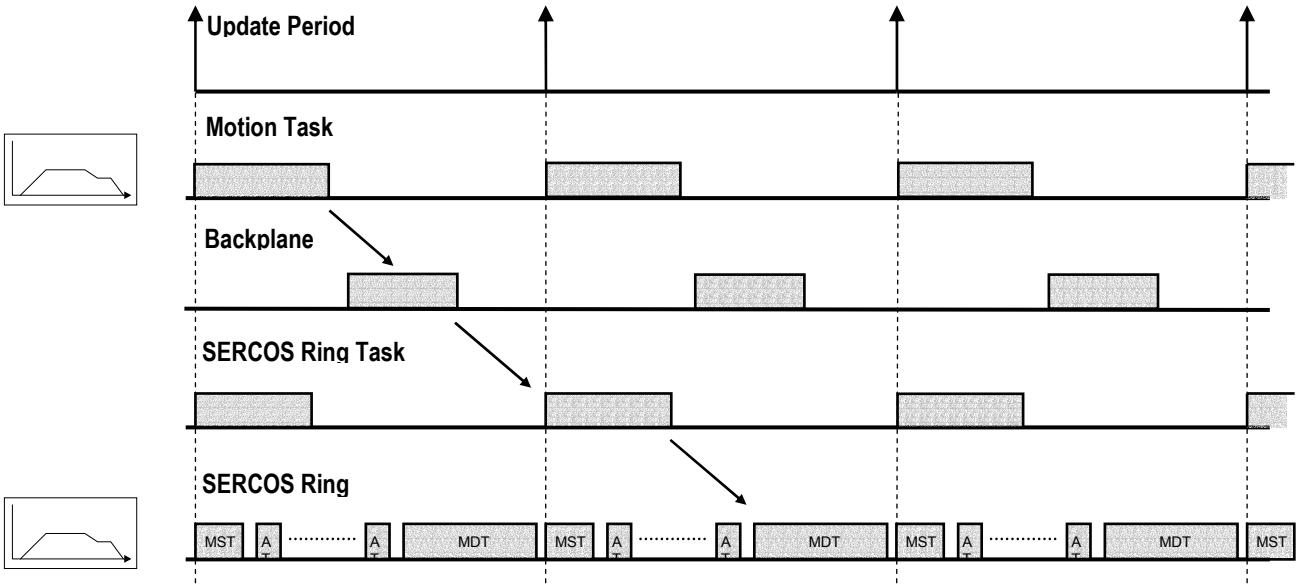



**Ethernet Adapter**

## 4.0 Motion Synchronization

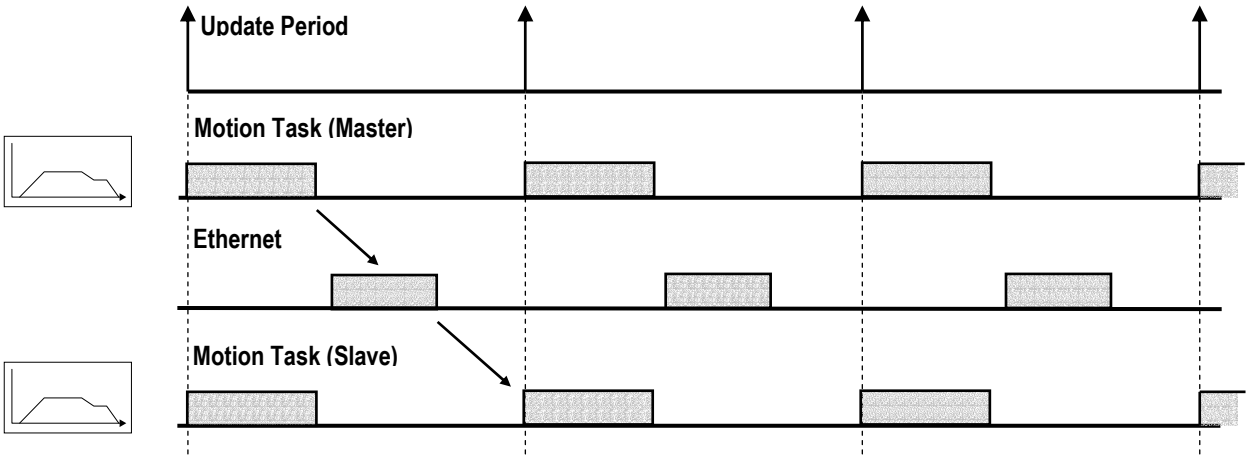
The basic motion operation requires the motion tasks running in each node to be synchronized to each other. Transactions between nodes are based on a synchronized periodic update cycle. This applies to both controller to drive transactions and controller to controller transactions.

For controller to drive transactions, at the beginning of the cycle the controller sends interpolated position updates to each of the drives. The drives use the position updates to control the closed loop position and velocity of the motor. Each drive returns its actual position to the controller. The controller computes a new position and the cycle repeats. This constitutes a position update cycle.



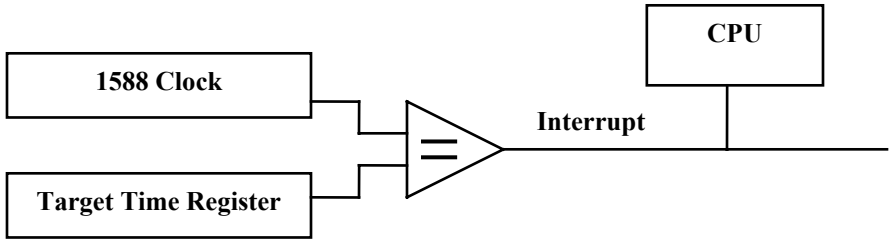

**Controller to Drive Transactions**

For controller to controller transactions, at the beginning of the cycle, the controller of the master axis sends a position reference to each of the controllers of the slave axes. The controllers of each of the slave axes use this position reference to "plan" the motion for the slave axis.




**Controller to Controller Transactions**

To synchronize all motion in the system, the motion tasks and consequently the position update cycles are synchronized to the 1588 clock. A small circuit in the FPGA is used to provide a periodic interrupt to the CPU to trigger the position update cycle. The circuit compares a time which has been loaded into a target register with the current 1588 clock time. When the current time matches the target time an interrupt is generated. In the interrupt service routine the CPU then loads a new target time equal to the current target time plus the cycle period and the process repeats. The phase and period of the cycle are setup during the node configuration process.




**Task Synchronizing Circuit**

## 5.0 1588 Implementation

The 1588 protocol is a C/C++ implementation running on the adapter. Most of the 1588 protocol is implemented including Sync, Follow-up, Delay Request, Delay Response, and Management messages.

The 1588 burst protocol is used by the time slaves to speed clock synchronization during startup. A burst of eight Sync messages is implemented.

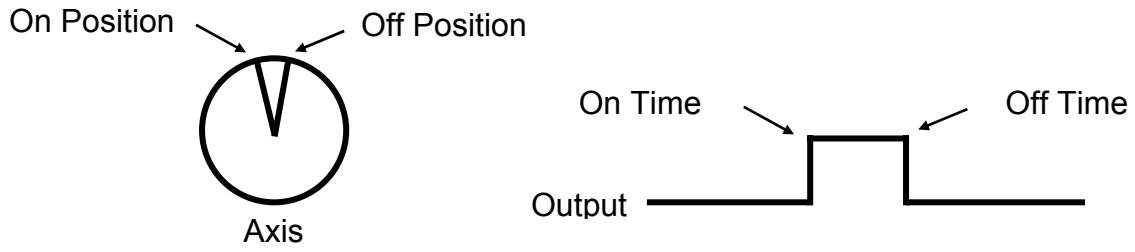
The “best master” algorithm is not implemented. Instead the system uses the “preferred” master selection to determine the time master for the subnet. On startup the slave clocks listen indefinitely for a master clock. The slaves never assume mastership. There is no provision for more than one “preferred” master.

Some support is provided to monitor the integrity of the time master clock. If a slave clock detects the loss of a master clock it stops its backplane clock. This causes the SERCOS adapter to shutdown the SERCOS ring and all motion stops.

## 6.0 Synchronized Outputs

In the prototype application there is a need to precisely turn an output on and off based on the position of the master axis. This output is used to trigger a strobe light to illuminate the phase position of all three axes. To achieve a precise output strobe a special output module is used whose clock is synchronized to the rest of the clocks in the system. An output value is sent to the module by the motion planner in the controller with a timestamp indicating the time at which the

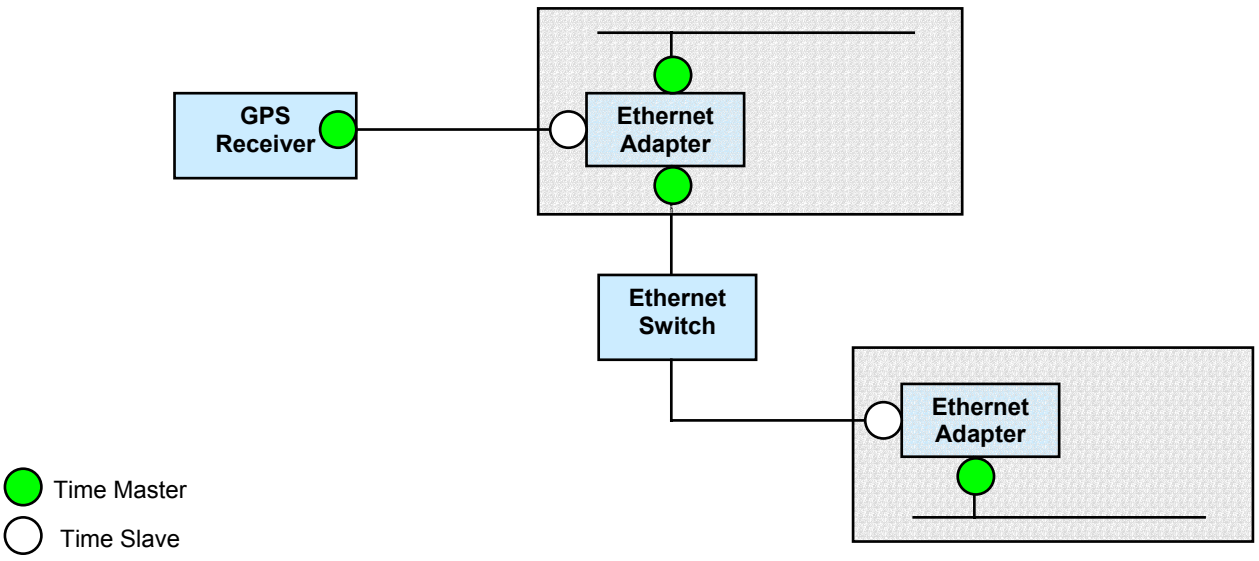
output should be asserted or de-asserted. The output module manages the output "schedule" using the task synchronizing circuit previously described to achieve precise output timing.




**Synchronized Outputs**

## 7.0 GPS as a GrandMaster Clock

The motion system prototype startup time defaults to a UTC time of 0. Absolute time is not generally needed for motion control, but can be useful for time stamping significant events such as fault conditions. A Global Positioning System (GPS) interface was implemented to provide an accurate source for UTC time and to serve as the GrandMaster clock for the rest of the system. This interface was implemented on the Ethernet Adapter module. An algorithm on the adapter receives "pulse-per-second" and UTC updates from the GPS receiver and makes adjustments to its local clock to maintain synchronization.




**GPS as GrandMaster Clock**

## 8.0 Results

The prototype application of 1588 with distributed motion over Ethernet proved to be reliable and accurate. The hardware assist circuit provides jitter accuracy under 200 nanoseconds between the master and slave clocks. When using GPS as the master reference, an accumulated jitter of 500 nanoseconds at the slave clocks results. The additional jitter is attributed to not having a clean edge on the pulse-per-second signal coming from the GPS receiver.

The prototype represents a relatively small system. Additional prototyping and testing are required with a more extended system under various load conditions.

## 9.0 References

1. IEEE 1588, Standard for a precision Clock Synchronization Protocol for Networked Measurement and Control Systems, 2002.